Forensic Science International: Reports 3 (2021) 100188

Contents lists available at ScienceDirect

Forensic Science International: Reports

journal homepage: www.elsevier.com/locate/fsir

Digital Forensics

A forensically-sound methodology for advanced data acquisition from embedded devices at-scene

Ben Findlay

School of Health and Life Sciences, Teesside University, Middlesbrough TS13BX, UK

ARTICLE INFO

Keywords:
Embedded devices
Router forensics
Digital forensics
Crime scene
Methodology

ABSTRACT

This paper presents a methodology for advanced extraction of data from embedded devices such as Internet routers. The use of different access techniques are considered, in order to gain access to device memory memory, and an investigative methodology is proposed. Lessons learnt from "hardware hacking" are considered and presented. Preparatory steps are discussed to maximise efficiency and likelihood of success for data acquisition. At scene actions and practice are discussed. The results show that, under the right circumstances, a full 'file-system' and a full 'physical' acquisition of the device's internal flash memory can be achieved. That this data can be decoded and extracted into a format which may subsequently be examined in industry-standard digital forensic tools is also presented and explored.

1. Introduction

The world of technology is ever-changing. The Internet has allowed for the use of millions, if not billions of connected devices by consumers, businesses, and industry. We exist in a world where almost anything and everything is now "smart" and "connected". Specialist, embedded devices exist, with dedicated functionality. These devices are common to almost all potential crime scenes, primarily in the form of an Internet router, but also in many other forms, often referred to as the *Internet-of-Things* (*IoT*). Such "embedded devices" often contain data storage which may be useful in a digital forensic investigation.

The International Telecommunication Union (ITU) defines *IoT* as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [1]. In Mukhopadhyay et al. [2], *IoT* is defined as "embedded devices (things) with Internet connectivity", and it is stated that "*IoT* has the potential to replace people as the largest consumer and producer of information". This is an ongoing process; the prediction of which been supported further by others. For instance, Miessler [3] notes and predicts a number of ways in which he believes *IoT* will change, and change how we as humans live. One key prediction Miessler makes is that data will become more open and readily available.

IoT has brought us a huge number of specialist embedded devices. An embedded device or system is defined by Heath [4] as "a microprocessor-based system that is built to control a function or range of functions and is not designed to be programmed by the end user in the same way that a PC is". However, the reality is that whilst they are not designed to be programmed by the end user; they must, by definition, be programmable in order for the developer and manufacturer to make them in the first place. The capability to do so is therefore obfuscated, but still present.

IoT can be loosely thought of as any specialist-function device which contains memory storage and communicates with other devices via a network or the Internet. Indeed, to quote the *Wired* [5] article *What is the Internet of Things?*; "from hairbrushes to scales, consumer and industrial devices are having chips inserted into them to collect and communicate data". These devices typically have a mainboard present, which runs an operating system. According to the Fraunhofer Institute [6] the vast majority of these are based on Linux.

The data these devices collect has the potential to be useful in ways the manufacturers and developers never imagined. From heart-rate data from a fitness smart-watch helping to support or indeed refute an alibi, to a mobile phone's GPS putting someone in a given location at a specific time, or even records from a home network router providing intelligence as to which devices are currently connected to it, these capabilities are varied and have significant potential to help solve crimes. All such devices must, by definition, contain some form of internal memory. It is common for users to interact with these devices through a specialist interface, application or, in the case of routers, via a web browser-based management console interface; and the devices themselves are generally designed in a way which prevents the user from directly accessing the physical memory storage.

Horsman, Findlay and James [7] propose a standardised method which can be employed by relatively unspecialised staff to extract meaningful intelligence and evidence from routers at crime scenes. It is therefore envisaged that this might be performed by a crime scene investigator and/or a digital media investigator, as opposed to a digital forensic specialist. The proposed method requires direct 'user-style'

https://doi.org/10.1016/j.fsir.2021.100188
Received 20 August 2020; Received in revised form 27 February 2021; Accepted 16 March 2021
Available online 22 March 2021

interaction with the device, and therefore it is to be expected that only an incomplete dataset is obtainable. This sort of acquisition is not carried out on all types of cases, but in fact could indeed be done as it is reliable, easy, and inexpensive to achieve. The method proposed by Horsman, Findlay and James [7] is designed to acquire data from running devices at scene, however it is important to consider the following questions and possible limitations of such a method:

- Does this method obtain all available data that can be obtained?
- What about volatile memory and/or data sources?
- What about historic logs that may not be accessible through the standard user interface?
- What happens to any such data when the device is removed from power during seizure?
- What happens in a situation where the device has had its default access password changed and the owner is not co-operative?

Horsman, Findlay and James [7] acknowledge such limitations; their research focuses on providing the front-line officer or investigator with a method to obtain meaningful data with minimal specialist training or skills. We should however consider methods to overcome such limitations, as access to greater amounts of data may be warranted, especially in high-risk, high-profile or major crime investigations. Lessons can certainly be learnt from the world of cyber security, specifically the "hardware hacking" community, in which exploitation of such devices is carried out in the interests of making them more secure. For digital forensic investigations, whilst the methods remain the same; the focus shifts from exploitation and compromise, to instead gaining access to devices with minimal footprint, for the ultimate goal of forensically sound and comprehensive data extraction.

Cusack and Lutui [8] identify the main challenge for the collection of data from routers, which they state to be the volatility of the data. They do not provide a solution for direct access to the memory. This is a problem which, as of yet, still has not been overcome. Clearly, for *IoT* to be used routinely in forensic investigations, a more standardised, established, reliable, and ultimately convenient method of access must be developed and implemented. Likewise, previous work in the field of cyber-security, such as that by Russell and Van Duren [9], has considered how these devices communicate as a potential method of obtaining data. The focus is almost exclusively on networking infrastructure, protocols, or "stack", rather than any physical access considerations. This situation clearly doesn't apply in a police-based investigation. Physical access, either at scene, or as a result of lawful seizure; followed by laboratory-based examination renders the possibility of physical access a near guarantee. Therefore, other methods can and should be considered.

The work by MacDermott, Baker and Shi [10] addresses challenges specifically associated with *IoT* devices, and concludes that current digital forensic methodologies are outdated. Indeed, the most recent copy of the *ACPO Good Practice Guide for Digital Evidence* was published in March 2012 (ACPO, 2012), and does not properly address the cloud, current mobile technology, or *IoT* devices. Their work also acknowledges that technology is changing the nature of how crime is committed. This is a view that is explored at length by Horsman [11], who proposes a series of updates to these principles. It is clear that the digital forensic industry must react and adapt to this fact.

The currently utilised standard methods for accessing router and embedded device data, such as either that advocated by Horsman, Findlay and James [7] or alternatively by means of dedicated specialist software and/or automated collection tools, typically rely on the investigator using the device in the same way as the end-user would; and capturing logically presented data through export, screenshot or other such method. Much of the underlying working of the device is therefore obfuscated and not accessible, meaning that significant amounts of data (particularly deleted and historic data) is inaccessible. This research will examine some techniques which may be utilised to obtain more data from such devices, using techniques which are typically associated with hardware exploitation and "hacking".

In order to carry out a digital forensic process, one must first acquire data to investigate. In conventional hard disc drive forensics, this is a straightforward and well-established process; typically utilising a write-blocker to produce an exact copy of the data stored on the storage medium. For mobile forensics, this process is more complicated, however a good degree of consistency and established processes still exist. For embedded devices such as routers and *IoT* devices this is even less straightforward; no standardised equipment or adapters exist which are universally recognised and can be used on a wide variety of models or equipment. Private companies and providers of forensic software, such as *MSAB*, *Cellebrite* and *HancomGMD*, do offer equipment with some such capabilities; however this equipment is often not universal (i.e. does not support a particularly wide array of hardware), and is typically expensive and therefore might be considered prohibitive for smaller service providers.

It is well established that the development of technology has historically put functionality before security. This can be seen in the work of Ngo, Nguyen, Le and Nguyen [12], which documents and discusses malware for *IoT* devices. For the digital forensic investigator, any such insecurity and flaws can actually be of benefit. Specifically considering routers; a study conducted by the Fraunhofer Institute [6] noted that none of the 127 routers they assessed were free of security flaws. Data held on storage media within emerging technology is often unsecured, not encrypted, and therefore accessible if physical access can be achieved. Until such time as the manufacturer implements security and privacy measures into their devices, the data is accessible and therefore relatively easily used in forensic investigations. Parallels of this are well documented in the arenas of mobile forensics and vehicle forensics.

Technology is becoming increasingly data rich. Common sources for such data include the administration console of the device (usually accessed via web browser from another device on the same network), a linked cloud storage account, and a dedicated 'app' installed on a smartphone. What is not currently typically considered for normal data acquisition is the device itself. Consider for example an investigation involving the Philips Hue system [13]. This system contains the following potential sources of data: a Philips cloud account, a smartphone/tablet app., the Bridge control unit which connects to a router, the router at the premises, and the smart bulbs.

The typical use cases and rationale for performing these different techniques are as follows:

1. Control of the scene/network: having a deeper access to the device (e.g. router) may enable access in situations where the wireless key or device admin access password is not known, and the owner is unwilling to provide it. These methods may also allow for establishment of the administrative passwords, determination of said passwords through other cryptographic means (e.g. cracking the hash), bypass of the security present, and changing of credentials in order to lock the suspect out and secure the network and/or devices from tampering
2. Access to produce a filesystem/physical acquisition of the device memory for subsequent "deep-dive" forensic analysis. Such acquisitions will typically include user-inaccessible data, such as deleted data and historic logs
3. Access to volatile data such as RAM on the device

Kaur and Kaur [14] further identify 4 key challenges in digital forensic investigations. The first of these is the diversity of different devices which may be encountered. This is a problem which has only been compounded by the recent development and proliferation of *IoT* devices. This is where examination of common hardware interfaces and designs may prove beneficial; by identifying common designs, interfaces and/or software, a solution may be proposed.

Any such investigative work must of course be undertaken with due respect to the regulatory frameworks in place within the jurisdiction. In the case of this research, UK law enforcement must comply with the law, the Association of Chief Police Officers guidelines [15] for digital

evidence, the ISO/IEC 17025 standard, and also the Forensic Science Regulator's (FSR) Codes of Practice and Conduct [16] where applicable. Furthermore, the FSR has indicated that ISO/IEC 17020 will also apply to digital forensics, however the details on this are still to emerge, and indeed the deadline of October 2020 has been deferred. International considerations must be made by those in different jurisdictions, however general concepts regarding forensic soundness, validity of evidence and sound scientific methods should all be transferrable.

Scientific frameworks exist for assessing the validity of specialist forensic work, such as the *Framework for Reliable Experimental Design* (FRED) [17]. In this work, Horsman states "To achieve this level of reliability, investigatory research must be suitably planned, implemented and analysed in a way which instills confidence in the accuracy of any findings". Similarly, in Horsman's Digital Evidence Reporting and Decision Support (DERDS) framework [18], it is highlighted that "it is crucial for any practitioner to possess the ability to make reliable investigative decisions which result in the reporting of credible evidence". The methodology proposed here is designed with these principles in mind, and to provide data compatible with sound, scientific principles of investigation. Indeed, what is discussed and proposed here with regard to access to the data fits within the *Acquisition* stage of the *General Procedural Framework* aspect of Horsman's model, and the subsequent analysis of any extracted data can be specifically undertaken in a manner which aligns to the *Artefact-Specific* aspect of said Framework.

On a practical level, thought must always be given to the validity of any potential data extraction technique or method. Given the devices in question are almost exclusively running a Unix/Linux operating system, in regards competency standards and accreditation, good familiarity with such systems is highly advised prior to attempting this work.

The aim of this methodology is to present a "forensically sound" procedure which can be utilised to obtain a greater amount of data from embedded devices than is currently the norm, and which can be achieved at scene in a safe, non-destructive manner. The methodology and techniques conducted here in this research are focused on router devices and the subsequent network forensics which may be conducted at-scene and highlighted relevant examples are provided using a piece of test equipment; however it should be noted that these methods and techniques have also been tested and shown to work on other embedded devices, including smart-home lighting equipment. The process of and responsibility for mapping this methodology into an existing Quality Management System or scope of accreditation sits with each organisation who intends to do this work going forward. The methodology provided herein will provide guidance and a framework for doing just that. It should also be anticipated that devices will be encountered for which this level of access is not possible; manufacturers may lock down such access or remove the connectors from the mainboards. This is simply a reality that must be accepted. Similarly, some connectors will be obvious, and others will take time and skill to identify; the Internet may provide a ready source of helpful information here. Identifying non-obvious electrical components which can be used for such connections is outside the scope of this research, and has been well documented elsewhere.

In order to effectively acquire data from embedded devices, certain skills and knowledge are required of the digital forensic investigator:

- A good general working knowledge of Linux operating systems and file systems
- Familiarity with running tools and commands via the command line in Linux; specifically, but not limited to *tar*, *dd*, *netcat*, *cat*, and *cd*.
- A basic understanding and knowledge of electronics, so that appropriate connections (such as UART, JTAG TAPs, GPIO etc.) can be identified and connected to. Detailed knowledge of the specifications and inner workings of such connections is not required.

In addition, the following specific software tools will be of use in such investigations:

- *Binwalk*; available from https://github.com/ReFirmLabs/binwalk
- *Squashfs-tools*; available from https://github.com/plougher/squashfs-tools
- *Sasquatch*; available from https://github.com/devttys0/sasquatch
- *Jefferson*; available from https://github.com/sviehb/jefferson
- *Ubi-reader*; available from https://github.com/jrspruitt/ubi_reader
- *Yaffshiv;* available from https://github.com/devttys0/yaffshiv
- *MTD-Utils*; available from http://www.linux-mtd.infradead.org/source.html

It is therefore advisable that the digital forensic investigator is familiar with these tools also. It should also be noted that these tools may be available in the inbuilt package repositories of the more mainstream Linux distributions, making installation more straightforward. Well-known cyber-security-oriented distributions, such as *Kali Linux*, and hardware-hacking-oriented distributions, such as *AttifyOS*; also include some of these tools pre-installed.

It is not anticipated that this methodology be carried out on each and every case involving an embedded device, only those where the content may be deemed relevant. Such examples include cases where the use of the device may provide an alibi or account for an individual being present at (or indeed absent from) a specific location at a certain time.

A final note for consideration is that this entire proposed methodology can of course be performed in the lab, in circumstances where the router or embedded device is seized. Whilst it is the case that some volatile data will be lost as a result of removing power from the device for transport, however it must also be acknowledged that these methods will still return more data than the currently established, user-based preview methods, and therefore they are still worth carrying out.

2. The proposed methodology

The proposed methodology for the investigation of embedded devices at scene is divided into 3 stages, as follows:

1. Pre-scene preparation
2. Peri-scene investigation
3. Post-scene digital forensics

2.1. Pre-scene

Prior to attending scene, if the devices that are present are able to be established in advance, a significant amount of useful research can be conducted in order to prepare and choose techniques and strategies to perform upon arrival. Useful information relating to the hardware present in embedded devices is often freely available via the Internet. There are a number of reliable sources for this information, including but not limited to the *FCCID* database [19], the *OpenWrt* Project [20], and *IFIXIT* teardown guides [21]. In addition to these, YouTube and Google also provide an excellent source of information in respect to hardware teardowns. A stockpile of common test devices could be maintained, so that first responders can liaise with digital forensic specialists in advance of them attending, enabling preparatory research to be conducted. Where there is little or no publicly available detail regarding a specific piece of device hardware, test equipment can always be purchased from retail suppliers in order for research to be conducted first-hand at short notice. Whilst it is true that these devices have in common that they almost all run on Linux, it must be recognised that different makes and models of devices will have differences in how and where key data is stored. Therefore it is vital that proper research and information gathering is conducted prior to investigating a live device at a crime scene.

Device mainboards often contain access points which are intended for developer use, but which can be exploited for digital forensic purposes. With many common embedded devices, it is possible to remove the outer cover and gain access to the mainboard whilst the device is running. The use of techniques such as *ChipOff* are well established in

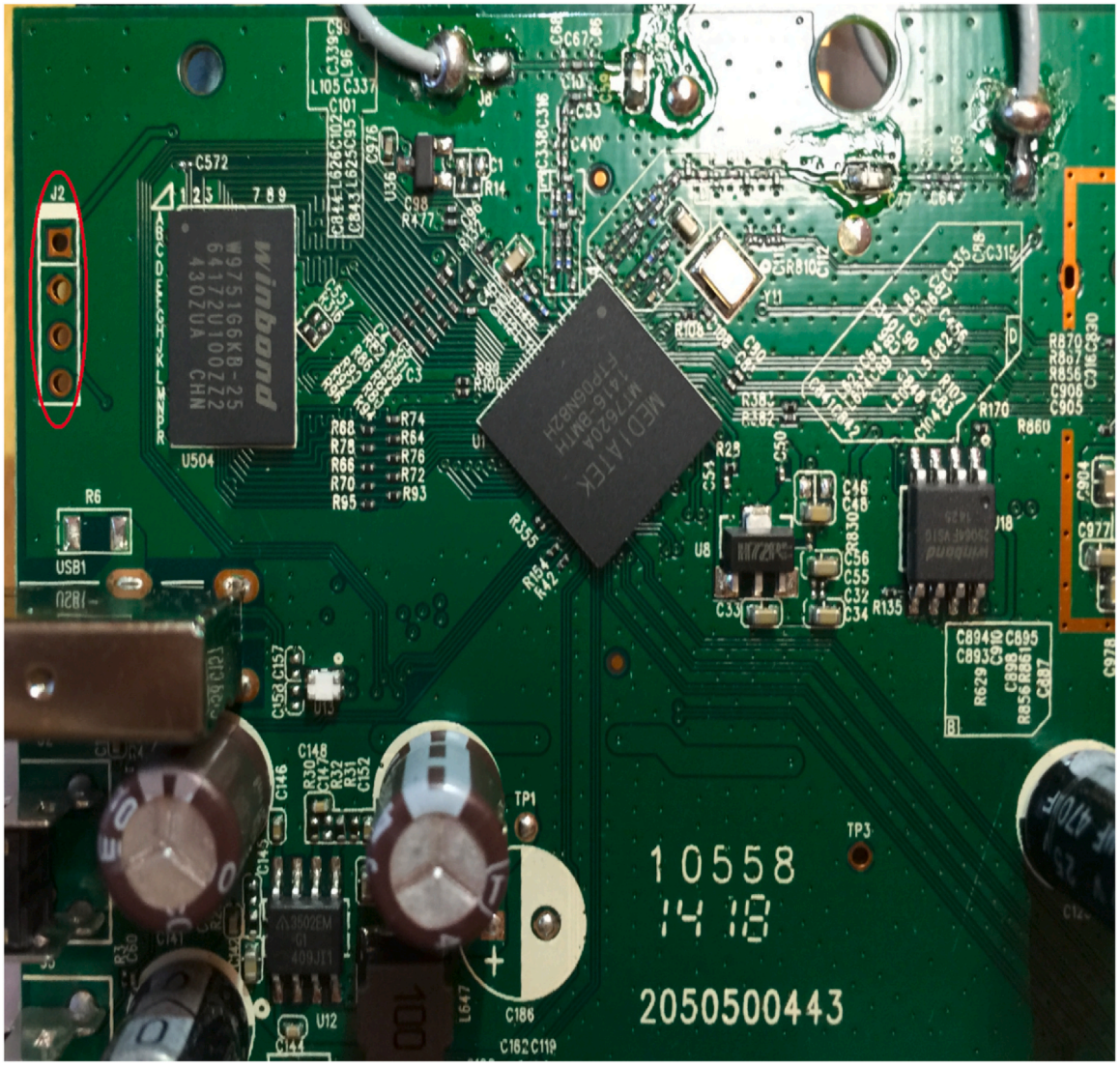

Fig. 1. Image of a router mainboard with UART (Serial) connector highlighted in red (top left). OpenWrt [23]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

lab-based digital forensics, however these techniques are destructive in nature and not feasible at-scene. Indeed such techniques do not need to be destructive; if the device has connections such as *JTAG*, *GPIO*, or *UART*, then these can be readily exploited to gain access to the data present on the device without damage or destruction occurring. These connection points may be obvious, as in the case shown in Fig. 1, however they may also not be immediately apparent, thus more detailed research may be required to locate them. In some cases, developers remove these connections once a device is programmed and configured. If this occurs, then it should be recognised that the proposed methodology and procedure becomes significantly more difficult and may in fact become impossible. As of writing, there are 1824 devices in the OpenWRT database [22], of which 400 are listed as having JTAG capabilities, 1467 as having Serial/UART capabilities, and 121 devices have some form of GPIO connection. Other methods of access may of course also be possible, such as *telnet*, *ssh* or *ftp* (if supported), however this data is not quantified here.

Consideration should also be given to obtaining information from the device manufacturer in two ways: firstly they may be willing to co-operate and provide much needed advice and guidance, and secondly their website may provide access to firmware (in the form of downloadable updates) which can be very useful. Firmware updates can be downloaded and deconstructed using specialist tools; such endeavours may often reveal hard-coded credentials such as admin passwords to access the device, which may be able to be used in addition to any default credentials which are printed on the back of the device itself. In at least one case in conducting this research, a plaintext password was extracted from a stock router firmware file obtained directly from the manufacturer which provided root access to the device in question, even though the default web admin password had been changed. In another example, the hash of the admin password was extracted from the firmware, which could then be subsequently cracked using password cracking software. Consideration should also be given to downloading multiple versions of old firmware; the author has observed passwords present in plaintext in older versions of firmware, which have been stored in a correctly obfuscated manner in newer versions of firmware for the same device. In this situation, the hash of the newer version was cracked and it was found that the password was still the same, however the point here is that older versions of the firmware may provide ready access to the password without the need to resort to cracking the hash in the newer version.

Utilisation of manufacturer-supplied firmware can be seen to comply with Horsman's FRED model [17], discussed earlier; as a method for safe, non-destructive testing; and to allow an appropriate investigate strategy to be designed, and good decision making to occur. The DERDS framework [18], also discussed previously; promotes effective decision making and advocates for thorough testing to be undertaken where required. In

the context of this research, the methodology proposed demonstrates precisely that; thorough, repeatable and (most importantly) useful research can be conducted in advance of attending a scene, which will ultimately lead to more effective acquisition of data.

To summarise the pre-scene stage:

1. Identify the devices present in advance of arrival if possible
2. Examination of publicly available information such as mainboard schematics/teardown guides may provide insight into connectivity to the device, such as *UART* or *JTAG* which can be exploited to gain access
3. Conduct research on test devices if available
4. Obtain firmware from the manufacturer. This can be used to reverse engineer data structures, file system information, passwords, and can provide meaningful intelligence on key files in advance. Examine multiple versions of firmware if available

2.2. Peri-scene

At scene, having identified a method of access, this may now be executed. Commonly available methods include making use of SSH access over the network, establishing a Serial connection, exploiting *JTAG TAPs*, and *SOIC/EEPROM* chip extraction.

In order to access the device via *JTAG*, *UART*, *GPIO* or other connection, it will of course be necessary to partially disassemble the device. This can generally be achieved whilst the device is still powered, and should be attempted this way, in order to minimise the loss of potentially volatile data. Disassembly is usually straightforward to achieve, and indeed, the preparatory actions undertaken during the pre-scene stage outlined earlier will give the investigator advance knowledge in order to do this quickly and effectively at-scene. At-scene this should be no more than the work of a few seconds.

Utilisation of *SSH* requires nothing more than a properly prepared forensic computer being present on the same network; a Terminal connection is established and login credentials are presented in order to gain access. The level of access gained through this method will depend on the account used to log in; if indeed any credentials are required.

Establishing a Serial connection will typically allow for root access to the device. A Serial cable is required to connect a properly prepared forensic computer to the device. These are typically very inexpensive. Some disassembly of the device is usually required for this. Linux/Unix has inbuilt Terminal support, Windows users can make use of tools such as PuTTY to achieve this. Hardware information printed to screen can be of use to an investigator, as this may provide live updates regarding networked devices connecting to and disconnecting from the device being investigated.

It should also be noted that a Serial connection is a useful source of diagnostic information when attempting and utilising other methods of access to such devices. It also provides meaningful information about the device, as can be seen in Fig. 2; where the boot sequence of an example router can be observed. This may be of use to an investigator as it often contains relevant information, such as details of filesystem and storage schema.

Fig. 3 shows hardware information being printed to the virtual console of said router, and confirms that a root shell has been obtained.

JTAG is another technique which can be utilised to gain access to the memory of embedded devices. The use of this technique in mobile forensics is well documented and will therefore not be explored further here.

SOIC can be used to interface with and download data from the *EEPROM* flash memory chips directly. It is used by developers to program chips and also commonly by “hardware hackers” to extract firmware from memory chips for vulnerability analysis. In a digital forensic context, this method may provide a physical image level of access to the device. Access to a logical file-system image will likely not be possible through this technique owing to the nature of how a connection is established. It is also the case, that live triage may not be possible, as this requires the establishment of a Terminal connection to the device whereas again, this method is typically used to directly access the flash memory.

A number of different connections have been discussed here. As previously discussed in the pre-scene stage of this research, appropriate connections can be determined prior to conducting the scene work, through research into the mainboard present within the device in question. This work focuses on the use of *UART* as an example, as that is what was present on the board of the test device shown here. *UART* also has the advantage of providing a *Terminal* style connection for live

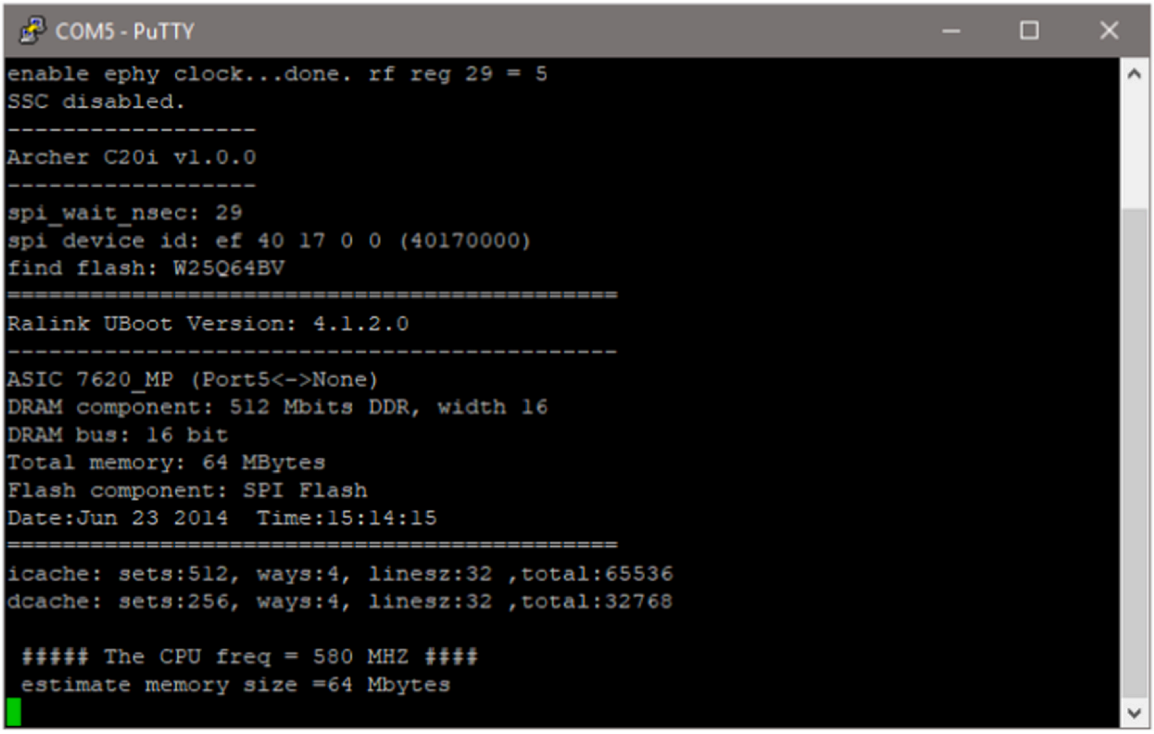

Fig. 2. Serial connection showing the boot sequence of a connected router.

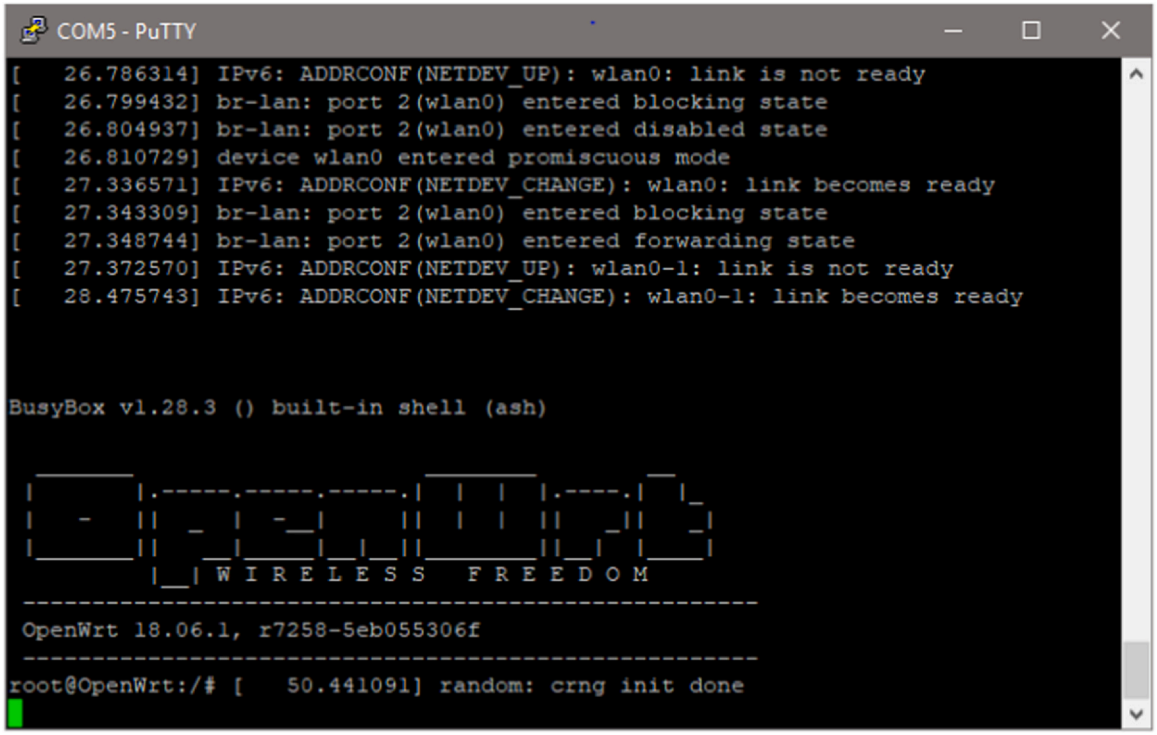

Fig. 3. Diagnostic information including state of networking hardware and a root shell.

interaction with the device via the command line interface. Once such a connection has been established, a number of different useful intelligence-based tasks can now be undertaken. These include:

- Basic filesystem navigation: the files and folders present can be explored and key files can be navigated to and examined, primarily through the use of commands such as *cat* or equivalent (see Fig. 5)
- Password extraction and/or reset: the *passwd* and *shadow* files can be extracted (via the filesystem navigation methods discussed immediately above) and sent for cracking with other tools. Alternatively the hashed credentials in the passwd and shadow files can be replaced with the hash of a known password, to enable access to the device. Or even more simply, it may be possible to use the inbuilt *passwd* command to replace the password with a known one manually. Replacing this password will then typically enable access to the web-based admin console in the situation where the owner of the device is not co-operative (see Fig. 6)
- Determination of the wireless password: commonly stored in non-volatile memory (in our example present in the file */etc/config/wireless*); this can trivially be obtained via Terminal using the *cat* command or equivalent. Fig. 4 shows the password for a wireless access point revealed on a running router via a Serial connection

The password-related steps discussed here are highly recommended, owing to the fact that suspects may re-use passwords. If for no other reason than it is possible that one of the passwords used here may be relevant to an encrypted file elsewhere in the case, these steps should be undertaken.

Fig. 6 shows the content of an example *passwd* and *shadow* file, obtained from a running router through the use of the *cat* command and a Serial connection.

Enumeration of the flash memory and identification of the filesystem volume structures is possible through the Terminal (see Fig. 7), by exploration of the mount points. Here the */etc/fstab*, and */proc/mounts* files should be examined for useful data. Furthermore, the content of */proc/mtd* is likely to be relevant. This last file commonly contains the mount point "labels" and sizes the relevant blocks of flash memory, which provide an investigator with meaningful intelligence as to where the majority of data is likely stored, and which *MTDBlock* is likely to contain any information of relevance. From this research's own empiric testing, the *fstab* file provides limited value but is recommended to be examined nonetheless. The file at */proc/mounts* has been noted to provide information about the various mounted locations within the filesystem, in addition to the filesystem present within each. This file has also been observed to provide useful intelligence in relation to volatile memory, in instances where the embedded device's RAM is in fact just another part of the onboard flash memory.

Once the memory structures and filesystem information have been ascertained, it is possible to exfiltrate data. As these devices are typically Linux systems, standard tools such as *dd*, *netcat*, and *tar* are often present. These are all that is required to extract both a physical and filesystem image from the devices, using a properly prepared forensic computer connected to the same network as the device which is being investigated. Usage is typically as follows:

- Having identified the mount locations from the */proc/mounts* file, the *tar* command can be used to pipe this data to a compressed archive via *netcat* to create a filesystem extraction. A separate extraction is required for each mount location identified. Fig. 8 below, shows an example *tar* extraction of the "root" directory of a running router.
- Having identified the flash memory structures by examining the */proc/mtd* file, the *dd* command can be used to pipe each *MTDBlock* file via *netcat* to physical image files on the forensic workstation. A separate extraction is required for each *MTDBlock* identified. Fig. 9 shows an example of extracted flash memory from a router (a *physical* image)

The process of extracting flash memory via *dd*, or the running filesystem via *tar*, is a quick procedure to undertake. Because embedded devices typically only have a very small amount of memory, this will be achievable at-scene in a short space of time; usually a few minutes at most.

Extraction of data via *dd* will of course provide what is most akin to the traditional "physical image" typically desired in more conventional, hard drive oriented digital forensics. It should be noted however that a physical image may not be the best solution in all cases. Certain data may be encrypted in the physical image, but ultimately be dynamically

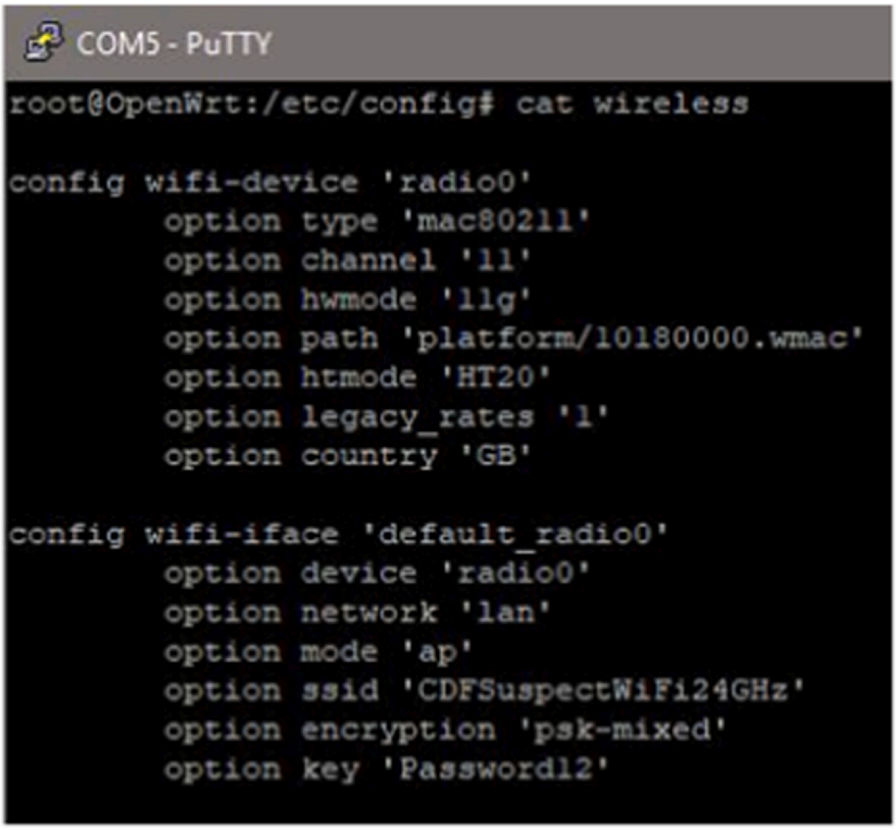

Fig. 4. Wireless SSID and password revealed through examination of the /etc/config/wireless file.

decrypted by virtue of the boot and/or other process. It is for this reason that it is recommended to obtain both the physical (MTD) and also the filesystem (*tar*) extractions as discussed; the *tar* image would be more likely to contain the data in a decrypted format, possibly removing encryption as a potential barrier to further investigation.

The forensic soundness of live examinations must be considered. It should therefore be noted that the majority of any such terminal-based interactions occur in RAM and therefore the footprint of changes on the running device is minimised. That being said, because the RAM may in fact be part of the onboard flash memory, care should be taken to run

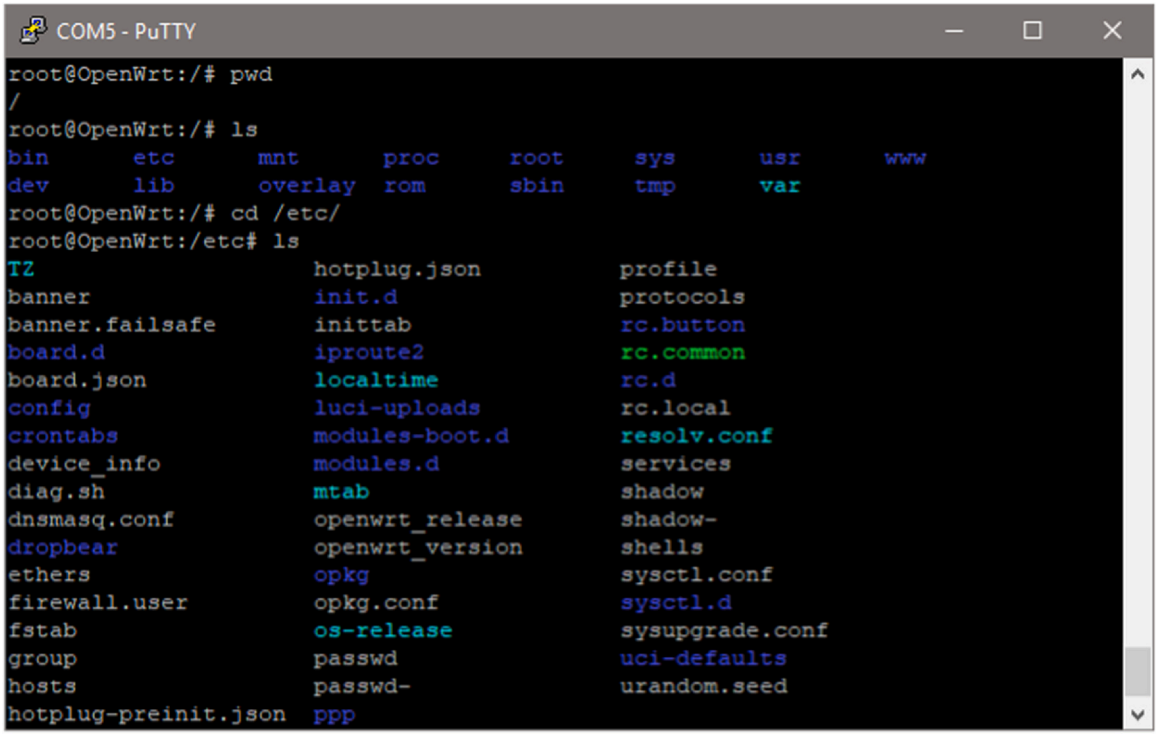

Fig. 5. File system navigation of the connected device.

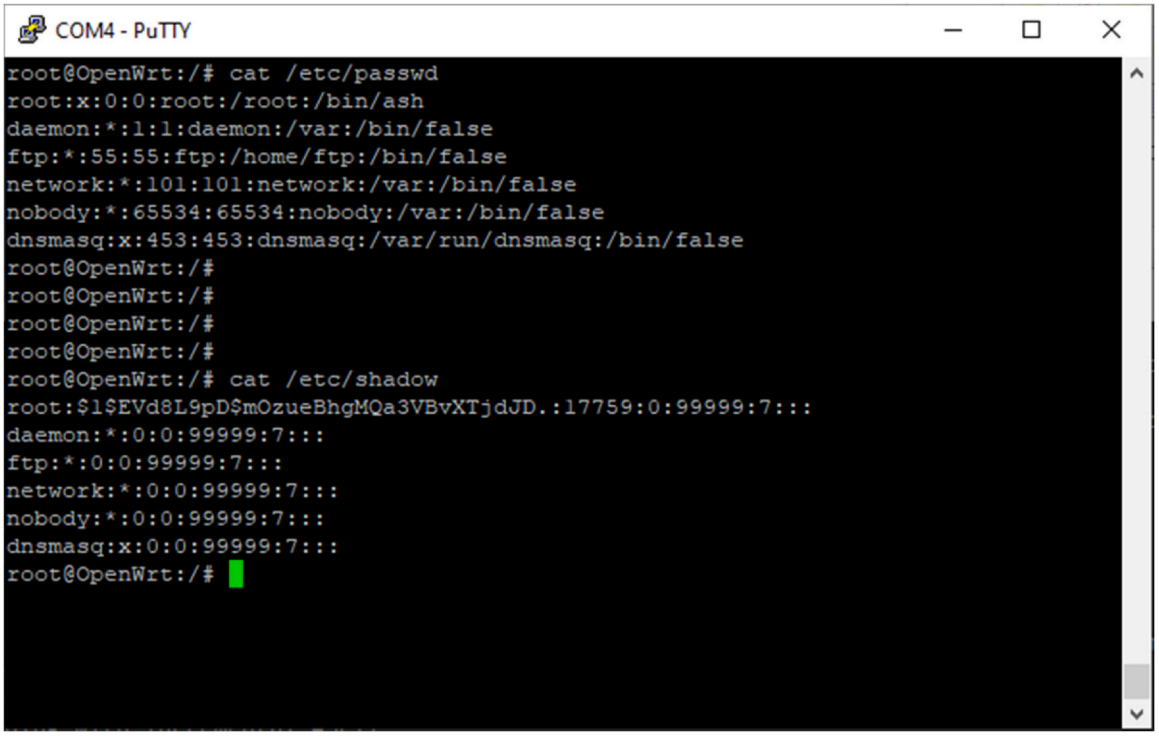

Fig. 6. The content of the *passwd* and *shadow* files, revealing password hashes.

the least number of commands as possible, in case these do result in the overwriting of previously existing data from within memory.

To consider some specifics of potential footprints which may be left behind, we must first consider the Linux operating system which, as previously established, accounts for the vast majority of operating systems present on embedded devices, such as routers. The following locations are examples of those which may result in a data footprint being recoverable as a result of this method of remote access:

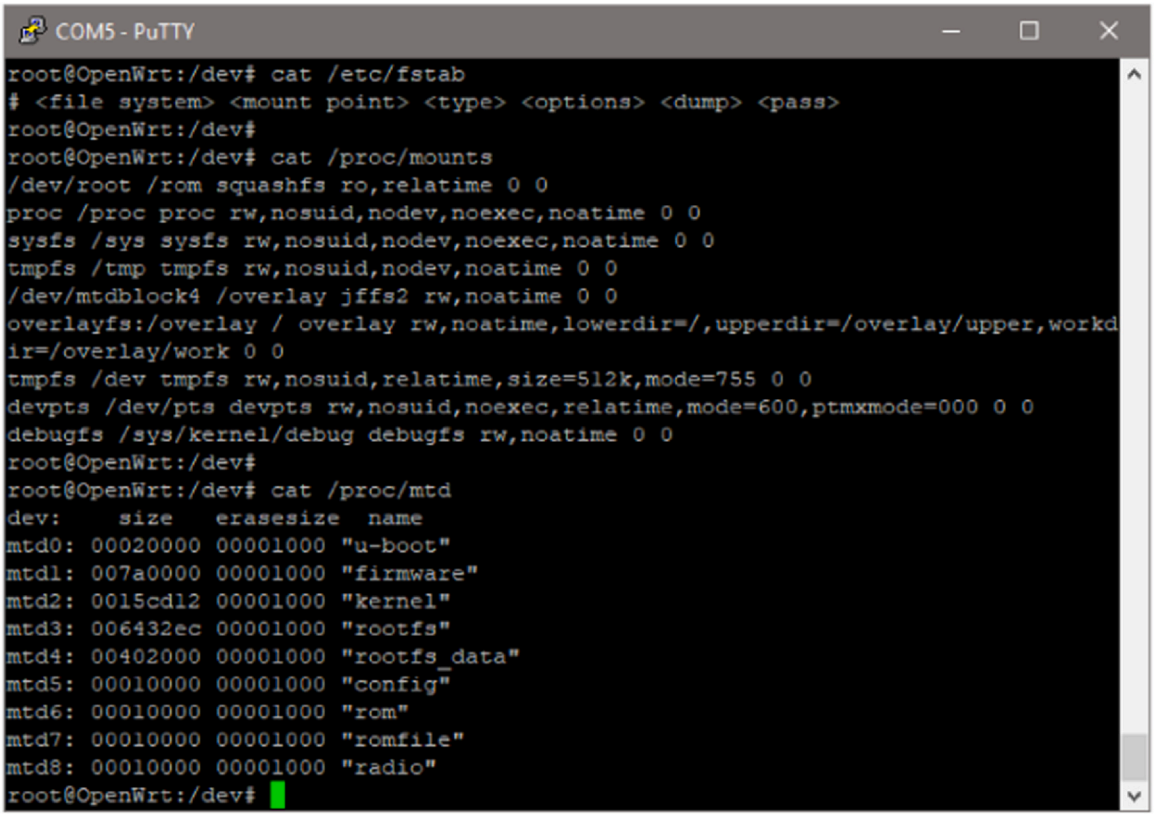

Fig. 7. Enumeration of mountpoints, filesystem volume information and flash memory structure.

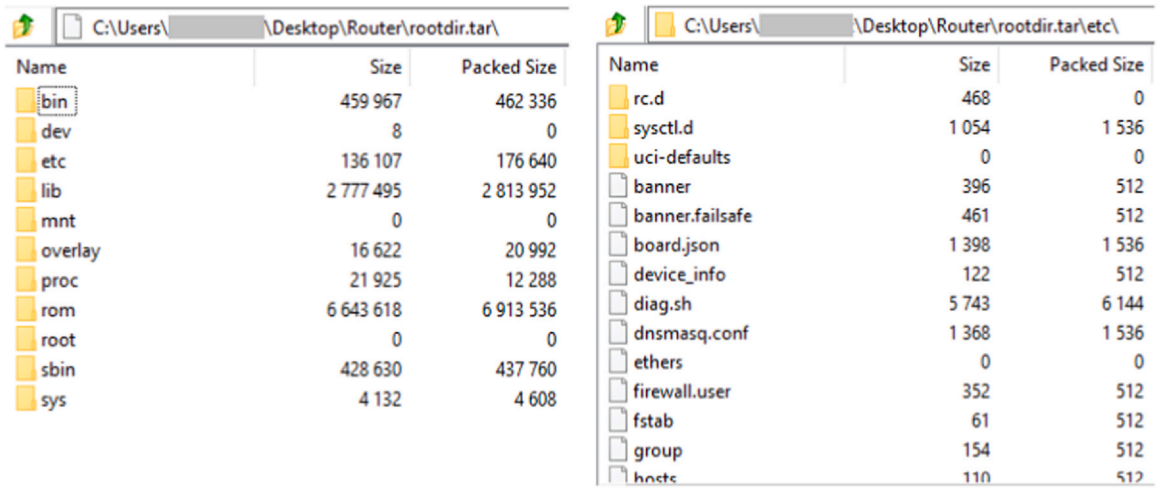

Fig. 8. "root" directory tar file, the content of the extracted data, and files present in the /etc folder present within this extraction, obtained from a running router.

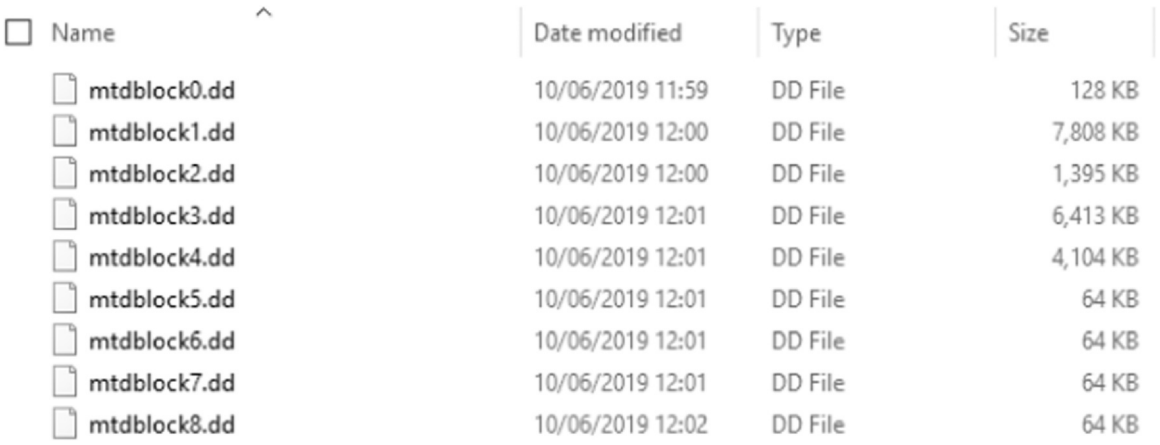

Fig. 9. Flash memory blocks obtained from a running router.

- Bash history. This is not normally updated as a result of a non-interactive or remote shell connection.
- Authentication logs; typically located under */var/log*. Should authentication be required, any failed authentications would be expected to be recorded in the *btmp* file, and successful logins under the *wtmp* file.

Clearly such interaction is a calculated risk, however it should be recognised that any such possible data footprints left behind by this style of connection and interaction would be considered acceptable in situations where remote imaging of a more "conventional" Linux system (such as a webserver) were required through similar or the same means (i.e. *dd* over *netcat*), and therefore could and should be considered acceptable here. It is recommended that further research can, and should, be conducted into this area.

Examination and extraction of data from running embedded devices naturally means that an investigator will be in close proximity to electrically live components. Therefore, precautions should of course be taken with regards to health and safety. Standard nitrile gloves do provide some protection against electricity; but as demonstrated by Deakin, Lee-Shrewsbury, Hogg and Petley [24] they do not withstand high voltages or currents. It should also be recognised that the above steps would ordinarily only be undertaken on equipment which appears to be in correct, and safe working order. Items at scene, which appear damaged or defective in any way should cause the investigator to first pause and assess their safety and stability to undergo any such further work. Should it be determined that the device is defective or unsafe in any way, consideration should be given to seizing the device for examination in a more controlled environment, such as a more fully equipped digital forensic laboratory. Proper diagnostic testing, performed by a qualified and experienced electronics engineer may be warranted. More involved diagnosis and invasive investigation, including the replacement of electronic components, such as is illustrated by the work of Heckmann, Markantonakis, Naccache, and Souvignet [25] may also be deemed proportional, especially in more serious crime investigations.

A final, but crucial point to consider during *peri-scene* operations is that a sufficiently knowledgeable and technically sophisticated suspect (i.e. one who has at minimum a sufficient level of knowledge to be able to carry out the very techniques discussed in this research) may have introduced counter-measures specifically designed to defeat such methods. A risk assessment should therefore be conducted prior to connecting to and interacting with the hardware under investigation at all. For the typical suspect, this will of course be a non-issue. In situations where the suspect is a "hardware hacker", or possesses significant knowledge and/or resources of a cyber-security nature (including, of course, organisational resources), then such techniques may be precluded. In this instance; more destructive approaches to extract the data (such as Chip-Off), or approaches which allow for the data to be extracted with the device powered off, are recommended.

To summarise the peri-scene stage:

1. Execute appropriate "hardware hacking" techniques to gain access to the device. Obtain a connection to the device using a developer style connection such as *UART/JTAG/GPIO*
2. Obtain intelligence by interacting with the device through an established Terminal connection. Credentials and key files can be examined through this interface and recorded in contemporaneous notes as appropriate

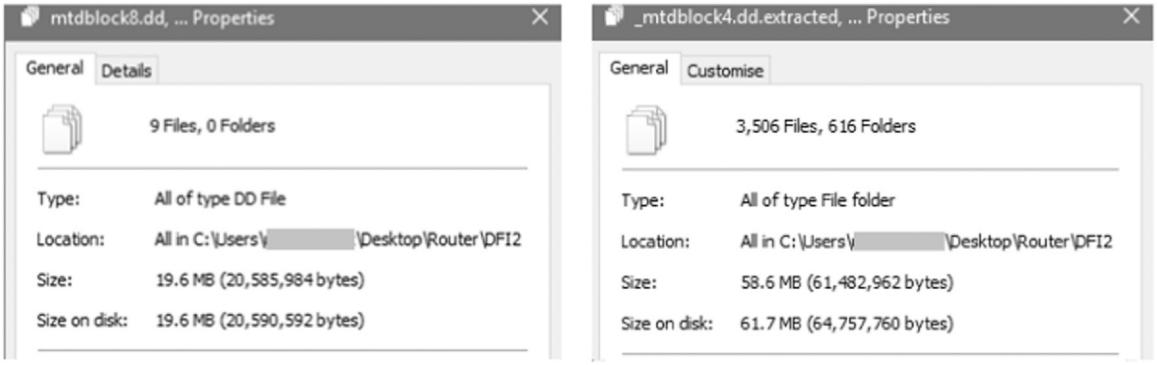

Fig. 10. Properties of the 9 MTDBlocks (left) and properties of the data extracted from within blocks 1–4 (right).

3. The device can be secured and control of the network obtained through the use of inbuilt commands within the running operating system
4. Export information out through appropriate channels (e.g. physical *dd* via *netcat*, filesystem acquisition using *tar* via *netcat*, EEPROM extraction, print and capture via terminal screen etc.). Performing both a filesystem and a physical image where available is highly recommended
5. Use as few commands as possible to further minimise an already small footprint

2.3. Post-scene

The investigator will now have either a collection of *tar* files containing the filesystem level extraction or a collection of *dd* files containing a physical image of the flash memory (or indeed both). The *tar* files are ready to examine as-is, although packaging into a logical evidence file or container format is highly recommended for the purposes of continuity and integrity.

As of writing, *MTDBlocks* and their highly compressed contained filesystems (typically *yaffs*, *squashfs*, and *ubifs*) are not supported by any of the mainstream digital forensic tools. Therefore it is necessary to first extract the data from these files, so that it can be analysed. Open source tools, such as *binwalk*, can be used in combination with other tools and plugins to achieve this. This will extract the data contained within the *MTDBlocks* into a subfolder and a folder structure and files will be presented which can be inspected logically. Subsequent addition of these extractions into a logical evidence file format, and proper preservation is highly recommended. Fig. 10 highlights just how much information is contained within these filesystems; on the left can be seen the properties of the 9 blocks highlighted earlier, and on the right can be seen the amount of extracted files and folders present in just the first 4 blocks. It should be observed that there are a significant number of files and folders present within the extracted content.

Once the blocks have had their data extracted, their content can be examined logically, as shown in Fig. 11 below. It should again be stressed that these files should be packaged into a logical evidence file or container format for the purposes of preserving the integrity of the data produced.

It is important to realise that there may be nothing of value present in certain memory blocks; indeed there simply may be no extractable

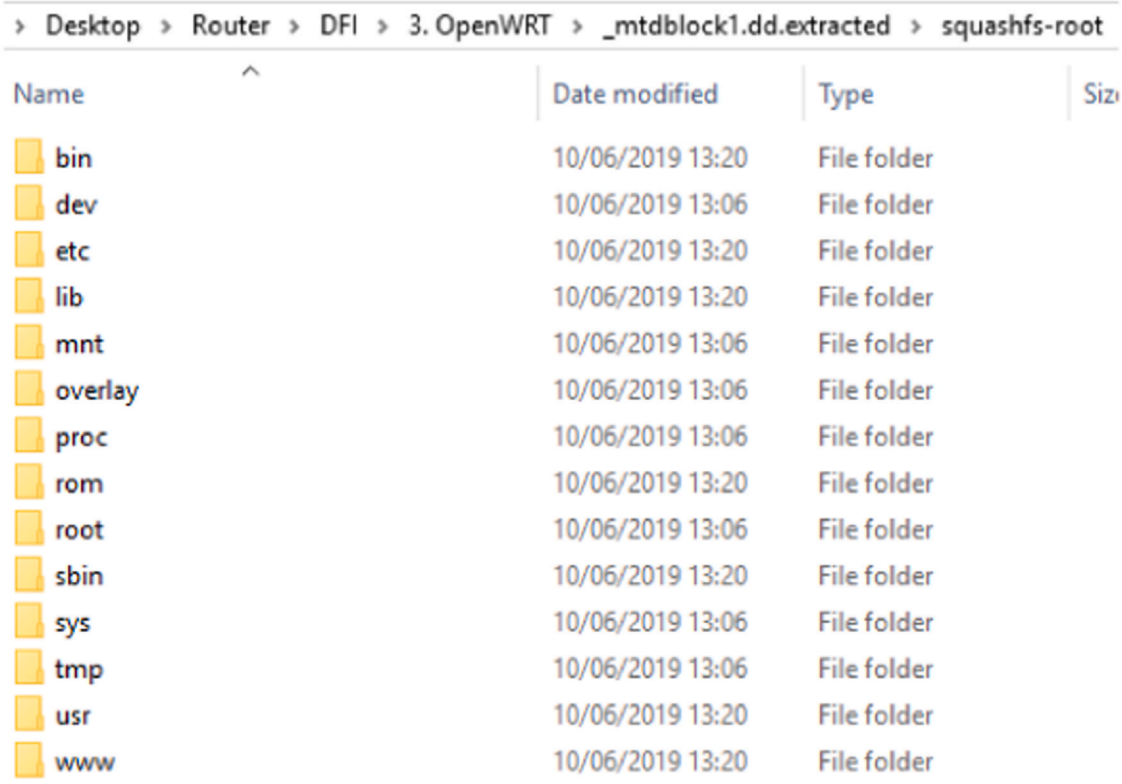

Fig. 11. Content of the squashfs filesystem extracted from MTDBlock1.

data present. Indeed, in this example the extraction tools were run across all 9 *MTDBlocks*, but only the first 4 produced any extractable material. The content of remaining blocks can and should always be examined manually in these instances.

To summarise the post-scene stage:

1. Extract data from acquisitions using relevant tools.
2. Extracted *tar* files should be packaged into a logical evidential format for preservation and to ensure integrity of the evidence
3. Extracted *dd* images of the *MTDBlocks* require processing and extraction with specialist tools to produce data which is compatible with the current digital forensic tool suites. This extracted data, like the *tar* files above, should be packaged into a logical evidential format for preservation and to ensure integrity of the evidence
4. Once the data is extracted and secured, it can be processed, analysed, and investigated using the digital forensic investigator's tool suite of choice

3. Conclusions

This research has proposed a methodology which can be utilised to increase the amount of data which can be extracted from embedded devices for use in digital forensic investigations. This methodology provides access to material which is normally inaccessible to the user and also to investigators using the current standard operating procedures. The benefits of pre-scene preparation, use of test devices and firmware exploitation have also been discussed, along with an explanation of methods to gain access and tools to use to extract data. These tools are all well-established standard issue operating system tools, which function in memory and provide a minimal footprint when in use, thereby increasing the validity, reliability and forensic soundness of their use. This research advocates use of the UART/Serial connection owing to its ease of use, reliability and inexpense. Furthermore it provides useful diagnostic information whilst performing actions, including when using it in combination with the other techniques discussed to obtain data. Finally, once the data is extracted, this methodology outlines the mechanism through which the data can be prepared for examination using standard digital forensic tools. To conclude, the proposed methodology has been provided in Appendix A for convenient reference.

Conflict of interest declaration

There are no conflicts of interest to declare.

Appendix A. SOP for advanced acquisition of data from embedded devices

1. Pre-scene:
   I. Identify the devices present in advance of arrival if possible
   II. Examination of publicly available information such as mainboard schematics/teardown guides may provide insight into connectivity to the device, such as *UART* or *JTAG* which can be exploited to gain access
   III. Conduct research on test devices if available
   IV. Obtain firmware from the manufacturer. This can be used to reverse engineer data structures, file system information, passwords, and can provide meaningful intelligence on key files in advance. Examine multiple versions of firmware if available
2. Peri-scene:
   I. Execute appropriate "hardware hacking" techniques to gain access to the device. Obtain a connection to the device using a developer style connection such as *UART/JTAG/GPIO*
   II. Obtain intelligence by interacting with the device through an established Terminal connection. Credentials and key files can be examined through this interface and recorded in contemporaneous notes as appropriate
   III. The device can be secured and control of the network obtained through the use of inbuilt commands within the running operating system
   IV. Export information out through appropriate channels (e.g. physical *dd* via *netcat*, filesystem acquisition using *tar* via *netcat*, EEPROM extraction, print and capture via terminal screen etc.). Performing both a filesystem and a physical image where available is highly recommended
   V. Use as few commands as possible to further minimise an already small footprint
3. Post-scene:
   I. Extract data from acquisitions using relevant tools.
   II. Extracted *tar* files should be packaged into a logical evidential format for preservation and to ensure integrity of the evidence
   III. Extracted *dd* images of the *MTDBlocks* require processing and extraction with specialist tools to produce data which is compatible with the current digital forensic tool suites. This extracted data, like the *tar* files above, should be packaged into a logical evidential format for preservation and to ensure integrity of the evidence
   IV. Once the data is extracted and secured, it can be processed, analysed, and investigated using the digital forensic investigator's tool suite of choice

References

[1] International Telecommunications Union. (2012). Overview of the Internet of things [ONLINE]. Available at ⟨https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=y.2060⟩ (Accessed 10 September 2018).

[2] S. Mukhopadhyay, Internet of Things, Springer International Publishing, Cham, 2014.

[3] D. Miessler, The Real Internet of Things, CreateSpace Independent Publishing Platform, 2017, p. 3.

[4] S. Heath, Embedded Systems Design, second ed., Newnes, Oxford, 2003.

[5] M. Burgess, 2018. What is the Internet of Things? WIRED explains [ONLINE]. Available at ⟨https://www.wired.co.uk/article/internet-of-things-what-is-explained-iot⟩ (Accessed 10 September 2018).

[6] Fraunhofer Institute. 2020. Home Router Security Report 2020. Available from ⟨https://www.fkie.fraunhofer.de/content/dam/fkie/de/documents/HomeRouter/HomeRouterSecurity_2020_Bericht.pdf⟩ (Accessed 19 August 2020).

[7] G. Horsman, B. Findlay, T. James, 'Developing a "router examination at scene" standard operating procedure for crime scene investigators in the United Kingdom', Digital Investig. 28 (2019) 152–162, https://doi.org/10.1016/j.diin.2019.01.010

[8] B. Cusack, and R. Lutui, 2013. Including Network Routers In Forensic Investigation. In Australian Digital Forensics Conference. Edith Cowan University, Perth, Western Australia, December 2013. Research Online: Proceedings of the 11th Australian Digital Forensics Conference. 58–70. DOI: 10.4225/7s5/57b3c682fb86d.

[9] B. Russell, D. Van Duren, Practical Internet of Things Security, Packt, Birmingham, 2016.

[10] A. MacDermott, T. Baker, Q. Shi, 2018. IoT Forensics: Challenges for the IoA Era. 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS). Available at ⟨http://researchonline.ljmu.ac.uk/id/eprint/7865/1/PID1206230.pdf⟩ (Accessed 19 August 2020).

[11] G. Horsman, 'ACPO principles for digital evidence: time for an update?', Forensic Sci. Int. Rep. 2 (2020) 100076, https://doi.org/10.1016/j.fsir.2020.100076

[12] Q.-D. Ngo, H.-T. Nguyen, V.-H. Le, D.-H. Nguyen, A survey of *IoT* malware and detection methods based on static features, ICT Express, Elsevier BV, 2020, https://doi.org/10.1016/j.icte.2020.04.005

[13] Philips. 2020. Philips Hue [online]. Available at ⟨https://www.philips-hue.com⟩ (Accessed 17 August 2020).
[14] R. Kaur, A. Kaur, 2012. Digital Forensics. International Journal of Computer Applications, vol. 50, no. 5 [ONLINE]. Available at ⟨https://research.ijcaonline.org/volume50/number5/pxc3880844.pdf⟩ (Accessed 10 September 2018).
[15] Association of Chief Police Officers of England, Wales and Northern Ireland. (2012). ACPO Good Practice Guide for Digital Evidence. Available from ⟨http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf⟩ (Accessed 18 August 2020).
[16] Forensic Science Regulator, 2020. Codes of Practice and Conduct for forensic science providers and practitioners in the Criminal Justice System. Available at: ⟨https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/880708/Codes_of_Practice_and_Conduct_-_Issue_5.pdf⟩ (Accessed 19 August 2020).
[17] G. Horsman, Framework for reliable experimental design (FRED): a research framework to ensure the dependable interpretation of digital data for digital forensics, Comput. Secur. 73 (2018) 294–306, https://doi.org/10.1016/j.cose.2017.11.009
[18] G. Horsman, 'Formalising investigative decision making in digital forensics: proposing the digital evidence reporting and decision support (DERDS) framework', Digital Investig. 28 (2019) 146–151, https://doi.org/10.1016/j.diin.2019.01.007
[19] FCCID. 2020. FCCID ID Search Tool. Available at ⟨https://fccid.io/⟩ (Accessed 17 August 2020).
[20] OpenWrt. 2020. The OpenWrt Project. Available at ⟨https://openwrt.org/⟩ (Accessed 17 August 2020).
[21] IFIXIT, 2020. IFIXIT Teardowns. Available at ⟨https://www.ifixit.com/Teardown⟩ (Accessed 17 August 2020).
[22] OpenWrt. (2021). The OpenWrt Project. Available at ⟨https://openwrt.org/toh/views/toh_extended_all⟩ (Accessed 2 Feb 2021).
[23] OpenWrt. 2020. The OpenWrt Project. Available at ⟨https://openwrt.org/_media/media/tplink/archerc20/archerc20i_mt.jpg⟩ (Accessed 18 August 2020).
[24] C. Deakin, V. Lee-Shrewsbury, K. Hogg, G. Petley, Do clinical examination gloves provide adequate electrical insulation for safe hands-on defibrillation? I: resistive properties of nitrile gloves, Resuscitation 84 (7) (2013) 895–899.
[25] T. Heckmann, K. Markantonakis, D. Naccache, T. Souvignet, Forensic smartphone analysis using adhesives: transplantation of package on package components, Digital Investig. 26 (2018) 29–39, https://doi.org/10.1016/j.diin.2018.05.005